

Grounding Description-Driven Dialogue State Trackers with Knowledge-Seeking Turns

Alexandru Coca[†], Bo-Hsiang Tseng[‡], Jinghong Chen[†], Weizhe Lin[†],
Weixuan Zhang[†], Tisha Anders^{†*}, Bill Byrne[†]

[†]Department of Engineering, University of Cambridge, United Kingdom

[‡]Apple

[†]{ac2123, jc2124, wl356, wz315, wjb31}@cam.ac.uk

[‡]bohsiang_tseng@apple.com *anderstisha@gmail.com

Abstract

Schema-guided dialogue state trackers can generalise to new domains without further training, yet they are sensitive to the writing style of the schemata. Augmenting the training set with human or synthetic schema paraphrases improves the model robustness to these variations but can be either costly or difficult to control. We propose to circumvent these issues by grounding the state tracking model in knowledge-seeking turns collected from the dialogue corpus as well as the schema. Including these turns in prompts during finetuning and inference leads to marked improvements in model robustness, as demonstrated by large average joint goal accuracy and schema sensitivity improvements on SGD and SGD-X¹.

1 Introduction

Task-oriented dialogue (TOD) agents provide natural language interfaces that users can interact with to access a wide variety of services, from airline search (Seneff and Polifroni, 2000) to complex customer service applications (Chen et al., 2021). To enable this, agents track key information communicated by the user as the conversation progresses. This is known as *dialogue state tracking* (DST). Commonly, the dialogue state is represented as a sequence of task-specific *slot-value pairs*².

A common DST assumption is that the set of slots and values a user may communicate, the *domain ontology*, is known at design time. Hence, extensive data collection and annotation is needed to support new domains, which hinders the scalability of this approach. Rastogi et al. (2020) address this issue by creating the schema-guided dialogue dataset (SGD). In SGD, the information available to a TOD agent is organised as *schemas*³ describing *services* with which users can interact. Each

service has *user intents* representing the tasks users can complete by interacting with the agent (e.g. *find restaurants*). Several slots are associated with each intent and the schema provides a *natural language description* for each intent and slot. The insight motivating *description-driven DST* is that these descriptions alone can be used in tracking the dialogue state in a form close to natural language. This has emerged as a powerful approach for few-shot and zero-shot DST (Jacqmin et al., 2022) and benefits from recent advances in language modelling. For example, Zhao et al. (2022) finetune T5 (Raffel et al., 2020) to generate the dialogue state conditioned on the dialogue history and a descriptive prompt containing all intent and slot descriptions in a service schema. The use of natural language in the descriptive prompt enables the underlying language model to generalise to new services, whose schemas are not seen in training.

While the reliance on natural language is a strength, Lee et al. (2022a) show that state-of-the-art (SOTA) schema-guided DST models are not robust to the style of descriptive prompts: in SGD, the training schema contains a single description per slot or intent, and models trained with prompts composed from this schema alone are prone to overfitting. Lee et al. (2022a) show this limitation can be mitigated by increasing prompt diversity. They use a large number of human annotators alongside expert curation to create diverse schema paraphrases that are used for model robustness improvement. This is a costly process that is not easily scalable.

As an alternative to additional human annotation of the schema, we show that the SGD training dialogues themselves exhibit sufficient diversity of expression such that they can be used to overcome the lack of diversity in the SGD schema descriptions. We *ground* DST prompts in the dialogue context by concatenating the schema descriptions with dialogue turns extracted from the SGD cor-

¹Our code will be released upon publication.

²For example, for a restaurant booking a sequence could be *day=friday, time=7pm, guests=1, restaurant=nandos*.

³See schema examples here: <https://bit.ly/3RJ6u4l>.

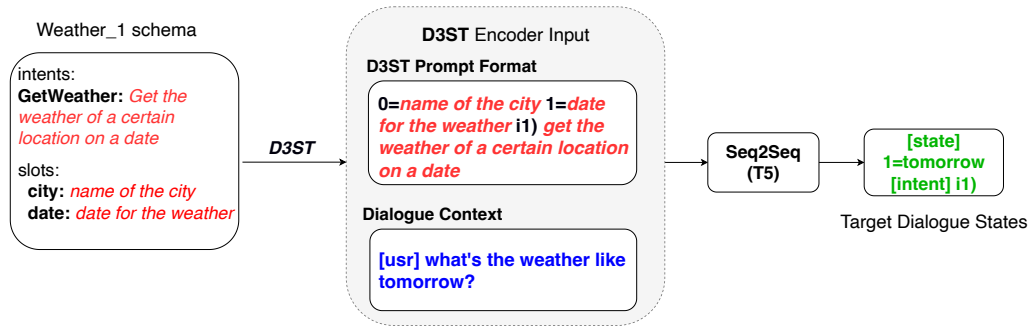


Figure 1: D3ST input and target format. On the left, we show a schema excerpt, where **slot** and **intent** names, in bold face, are followed by their *natural language description*. The encoder input, represented in the centre, is a string, the concatenation of two elements: the *prompt* which describes what information should be output and the *dialogue context* from which the information should be extracted. On the right we show the target dialogue state.

pus based on similarity to the dialogue state. We find that this approach is more effective than using synthesised prompts and even outperforms or is comparable to the highly-curated human-written prompts used by Lee et al. (2022a), when evaluated with medium and large language models. We evaluate our methods using the SOTA D3ST prompting scheme (Zhao et al., 2022) on the SGD and SGD-X (Lee et al., 2022a) datasets.

2 Related work

Neural classification is effective for DST when the domain ontology is fixed and known at design time (Mrksic et al., 2017). Adapting such models to track new slots and domains requires annotated conversational data and thus data scarcity is a long-standing issue in DST research (Jacqmin et al., 2022). Scarcity has been addressed by copy-enhanced generation (Wu et al., 2019), reading comprehension (Gao et al., 2019) and adapting pretrained language models to generate the state given the dialogue context alone (Peng et al., 2020; Hosseini-Asl et al., 2020). These were improved upon by transfer learning from question-answering tasks (Lin et al., 2021a), which in turn was outperformed by schema-guided models (Lee et al., 2021; Zhao et al., 2022; Lin et al., 2021b). Recently, Gupta et al. (2022) apply in-context tuning (Min et al., 2022) to DST, creating training prompts which contain a dialogue and its target state sequence. Their model thus learns from DST task demonstrations.

Lee et al. (2022a) investigate the robustness of SOTA schema-guided dialogue state trackers to schema changes. This is a new line of research, as previous work concerns other robustness issues that generally affect DST, such as variations in the

conversational data distribution, noise, and adversarial perturbations (Jacqmin et al., 2022). Through extensive, crowdsourced⁴, schema paraphrase collection, Lee et al. (2022a) report that DST performance degrades substantially when models trained on one set of prompts are evaluated on manually paraphrased prompts. By contrast, Cao and Zhang (2021) report little degradation in DST with backtranslated prompts, suggesting that backtranslation is a weak proxy for actual human variability. Lee et al. (2022a) perform data augmentation (DA) for robust DST, finding that prompts obtained via automatic paraphrasing lag in quality relative to manual paraphrases. Ours is the first work to address the gap between synthetic methods, such as backtranslation, and manual paraphrasing. We show that the gains from manual paraphrasing can be achieved by mining the existing annotated dialogues used for training the DST model in the first place.

3 Robust DST with grounded prompts

We review D3ST, a SOTA description-driven DST model (Section 3.1). We then describe our grounding method that extracts turns from the corpus (Section 3.2) and uses them to design prompts for robust DST with D3ST (Sections 3.3 & 3.4).

3.1 Description-driven dialogue state tracking

Figure 1 shows the inputs and outputs of D3ST (Zhao et al., 2022). The model is implemented with T5, an encoder-decoder language model (Raffel et al., 2020). The encoder input, represented in the centre, comprises a *prompt* describing what information should be tracked by the DST model, and the *dialogue context*, a conversation between

⁴The SGD schema were rewritten by over 400 annotators and curated by dialogue experts, over the course of one month.

a user and an agent from which slot-value pairs should be extracted. The prompt is a concatenation of slot and intent descriptions, extracted from the service schema (on the left). Each description is prefixed by a randomly assigned index prior to concatenation. Zhao et al. (2022) motivate their use of random indices to replace slot and intent names because names convey little semantic information and may be ambiguous⁵. In this paper, we will refer to this *prompt format* as **D3ST**.

The model is trained to output index-value pairs for all slots mentioned in the conversation (i.e. the *active slots*) as well as an index representing the active intent. These are represented on the right in Figure 1. The slot-value pairs mentioned in the conversation can be recovered by replacing the predicted indices with their corresponding slot names. The user active intent is found by replacing the predicted index with the name of the intent.

3.2 Mining turns for prompt design

1. REQUEST(<i>restaurant_name</i>) SYS: Where do you want to dine?
2. INFORM(<i>restaurant_name=Nandos</i>) USR: I want Nando's.
3. INFORM_INTENT(<i>find event</i>) USR: What shows are on?
4. OFFER_INTENT(<i>buy ticket</i>) SYS: Want tickets?

Table 1: Sample semantic annotations

We now discuss how to extract turns from the corpus to design better prompts. Our approach involves an automatic step that uses the semantic annotations in the corpus followed by a verification step to ensure that the turns selected are diverse.

Each turn in SGD is semantically annotated with one or more *dialogue actions* which describe what is being communicated (Table 1). We focus on *knowledge-seeking* turns (KSTs). These are annotated with a single REQUEST *dialogue act* and associated with a *single* slot, without a value mention (Table 1, line 1). Selecting turns annotated with a single slot allows us to unambiguously associate them with slots in the D3ST prompt. We do not mine *informational* turns (labelled with INFORM) since these mention a specific, known value, often without reference to the underlying slot (Table 1, line 2). Such turns could be combined with schema information to form *exemplar-based* prompts as

⁵For example, the *location* slot name may be used to refer to both a city name and an address across different services.

done by (Figure 1 in Gupta et al. (2022)), a more complex approach which we discuss in Section 5.5.

To select turns for a given slot, s , we filter the corpus to get all the knowledge-seeking turns relating to it. We manually select 5 of these, repeating this process for each slot in every service in the training data. See examples in Table 2. In a similar fashion, we select 5 turns from those labelled with a single INFORM_INTENT or OFFER_INTENT act (Table 1) for every intent in the training schema.

Index	Selected Knowledge-seeking Turn
1	Which event are you looking to book
2	Do you have any particular show in mind
3	And what is the event
4	What event do you wish to see
5	What is the event you are looking for

Table 2: Selected turns of the *event name* slot

We opt to select the turns manually because our goal is prompt diversity. Our SGD analysis revealed that the knowledge-seeking turns tend to be biased towards specific vocabulary and syntactic patterns. For example, among the 173 turns in which the user requests the price of a rental car, 71.1% contain the word *cost*, 42.8% contain the word *total* and, 27.7% contain *total cost*. In contrast, *price* appears in just 11.0% of the turns.

All turns were mined by one student in one day, despite SGD being the largest schema-guided TOD corpus. In practice, schemas are induced by developers from unlabelled conversation databases (Yu et al., 2022). The turns could be collected as part of this process with negligible overhead. We handle slots with few KSTs as described in Appendix A.

3.3 Grounding prompts

To ground a schema description in its conversational use, we concatenate it with randomly selected knowledge-seeking turns from the mined collection (Figure 2). In the example shown, the sampled knowledge-seeking turns for the *city* and *date* slots are *In which location should I check?* and *Forecast for when?*, respectively. These are concatenated with the original SGD schema descriptions *name of the city* and *date for the weather* to ground the prompt. We concatenate the turns and descriptions in random order, to prevent the model learning to attend preferentially to one source of information over another. We refer to D3ST trained with prompts grounded in knowledge-seeking turns as **D3ST-Turn** in what follows.

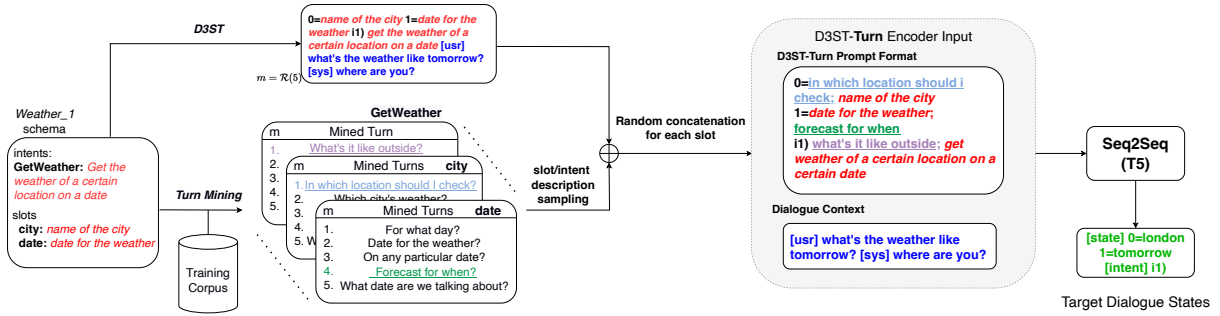


Figure 2: Visual representation of D3ST-Turn prompting. Underlined knowledge seeking turns are those chosen at random for inclusion in the sample D3ST-Turn prompt shown.

Slot names may provide additional information about the meaning of a slot, so we propose to ground the prompt both in knowledge-seeking turns and slot names. We refer to D3ST trained with prompts grounded in knowledge-seeking turns and slot names as **D3ST-TurnSlot**.

3.4 Grounded prompt ensembling

Multiple knowledge-seeking turns are available for decoding, enabling us to create multiple *prompt variants*. A given model generates the dialogue state when conditioned on each of these prompt variants, in turn. The state hypothesis is the most commonly predicted string when our *single* model is prompted with the prompt variants. We call this technique *grounded prompt ensembling* (GPE).

4 Experiments

4.1 Datasets and metrics

SGD (Rastogi et al., 2020) The training set contains 21, 106 dialogues across 16 domains. The test set contains 4, 201 conversations, 77% of which have a turn span where the user talks to the agent to access a service unseen in training. 6 schemas are seen in training whereas 15 are *unseen*. Hence, this benchmark primarily tests the ability of DST models to accommodate values, slots, prompts and domains it has not been trained on.

SGD-X Lee et al. (2022a) created SGD-X because they found the linguistic patterns of the SGD unseen services schemata to be too similar to those of the seen schemata⁶. They use crowdsourcing and dialogue experts to create five *schema variants*⁷ which are increasingly stylistically and lexically divergent from the SGD schema. A schema variant describes the same services as the SGD schema

⁶For example, descriptions of slots with "true" or "false" values always start with *Boolean flag indicating*.

⁷See examples here: <https://bit.ly/3Ev0KrV>.

but with increased linguistic variation. The variants are ordered by the Jaccard distance between the descriptions of the original SGD schemas and the schema variant descriptions. The *v1* variant is the closest while *v5* is the most dissimilar to SGD. Ideally, a robust model should output the correct state regardless of which schema variant is used for prompting.

Metrics Joint goal accuracy (JGA)⁸ is the percentage of turns where all the slot-value pairs from a given service are correctly predicted. On SGD, it is computed over seen and unseen services. The presence of the unseen services measures the ability of the DST model to make correct predictions for unseen slots and values and to interpret descriptions unseen at training time (Rastogi et al., 2020).

For SGD-X, we report the JGA broken down by seen and unseen services and their combination. The JGA coefficient of variation across the five schema variants is termed *sensitivity* (SS). It measures how well the model accommodates linguistic variation. Evaluation on the seen portion involves prompting with *paraphrases* of schemata seen in training. Performance decreases if the model overfits to the training descriptions. In evaluation on the unseen portion, the model is prompted with five distinct human-written prompts of increasing dissimilarity to the original SGD. This evaluates if generalisation is robust to linguistic variation.

4.2 DST models

Our baselines are three D3ST models⁹ trained with large augmented datasets. For every training example that uses the D3ST prompt format (Figure 1) linearised from the SGD schema, k additional training examples are created either using synthetic prompts or the $k = 5$ SGD-X schemata. We create

⁸We use the official evaluator: <https://bit.ly/3B7jD1c>

⁹We use T5-base (220M) for all models except in Sec. 5.6

augmented datasets using three methods, explained below. See Appendix B for implementation details.

1. Backtranslation We follow Lee et al. (2022a) to create $k = 3$ schema variants by backtranslating the SGD schema via Chinese, Japanese and Korean with Google Translate. The augmented dataset is 4 times larger than SGD (703, 120 examples).

2. Easy Data Augmentation (EDA) (Wei and Zou, 2019) We create $k = 5$ schema variants by applying word-level perturbation to the SGD schema (EDA). Synonym replacement is applied with probability 0.25 whereas random insertion, deletion and substitution are applied with probability 0.05. There are 1,054,680 training examples.

3. SGD-X We create 1,054,680 training examples using the $k = 5$ human-written SGD-X schemata. Unlike the other baselines, SGD-X-trained models see the human-written paraphrases of the seen test services during finetuning. In all other experiments, *none* of the SGD-X test prompts are seen during training, and so we refer to this experiment as an *oracle*, following Lee et al. (2022a).

Contemporaneous to our work, Coca et al. (2022) propose a tree-ranking approach for improving paraphrasing. While they show significant gains compared to the state-of-the-art backtranslation baseline we compare to in our work, we contribute to the body of knowledge on robust state tracking by showing novel augmentation and prompting techniques that achieve significant further improvements.

Grounded D3ST Instead of augmentation, we propose to ground D3ST in knowledge-seeking turns by finetuning T5 with the Turn (D3ST-Turn) and TurnSlot (D3ST-TurnSlot) prompts (Section 3.3) on a dataset containing 175,780 examples (SGD size). At decoding, the same turns grounding the training prompts are used for seen services. For unseen services, we select five turns per slot as described in Section 3.2. For each test example, we construct a prompt with the same format as in training using knowledge seeking turns selected at random, per slot, from the mined collection. This tests model’s ability to interpret additional task-relevant information.

5 Results and discussion

5.1 Robustness via data augmentation

Augmenting the finetuning dataset with the SGD-X prompts leads to a 13.2% improvement in D3ST SGD-X JGA (#1 vs #4, Table 3). In contrast, aug-

#	Model	SGD	SGD-X	Seen	Unseen	SS ↓
1	D3ST	69.8	56.5	73.6	50.8	70.1
2	D3ST + Backtrans. DA	72.1	62.2	84.0	54.9	53.1
3	D3ST + EDA DA	71.4	62.3	83.3	55.3	53.2
4	D3ST + SGD-X DA (oracle)	73.8	69.7	92.5	62.1	27.9
5	D3ST-Turn (ours)	75.9	69.5	88.5	63.2	36.6
6	D3ST-TurnSlot (ours)	74.7	72.0	90.7	65.6	23.7

Table 3: Grounded D3ST models outperform strong baselines in both JGA and SS. Seen and unseen numbers decompose the SGD-X JGA (Section 4.1), *oracle* indicates a model trained on SGD-X. "+" marks data augmentation (DA) during finetuning, and is followed by augmentation method name. Column maximum is in **bold**. In all tables, numbers are averages of three runs.

Schema	v1	v2	v3	v4	v5
Backtranslation	97.5	96.5	95.9	-	-
EDA	99.1	98.5	96.6	93.2	86.4
SGD-X	89.7	88.0	88.4	86.8	87.5

Table 4: Semantic similarity of SGD and schema variants, measured by entailment (Narayan et al., 2022)

mentation with synthetic prompts obtained through backtranslation or word-level augmentation improves performance by just 5.6%. On SGD, human-written prompts outperform the best performing synthetic ones (#2) by a margin of 1.7%.

Gains obtained with synthetic prompts reflect some degree of lexical and syntactic diversity in the generated paraphrases. For example, a backtranslation of *The amount of money to transfer* is *Amount to be remitted* and *The account type of the user* is backtranslated as *User’s account type*. The entailment scores (Table 4) show that backtranslation largely preserves the semantic content of the prompts. Meanwhile, if more edit operations are applied via EDA, the synthetic prompts are less faithful, as demonstrated by the sharp entailment decrease for the v4 & v5 variants. Robustness did not improve when we experimented with a larger backtranslation-augmented dataset (Appendix C).

The SGD-X schemata "do not fully semantically overlap with the input as traditional paraphrasing requires" (Lee et al., 2022a). This is consistent with SGD-X schema variants attaining lower entailment compared to backtranslated ones (Table 4). The annotators used the wider context of the service and common-sense knowledge to create diverse, high quality, schemas. Meanwhile, D3ST learns to identify slots using the uniform linguistic patterns of the SGD schema and it is not robust to the wide variety of styles annotators used in SGD-X. D3ST trained with augmented data via EDA or backtranslation improves compared to D3ST

trained on SGD alone, but the large performance gap to human-written prompts indicates that strict paraphrasing introduces less diverse, task-relevant, cues in the prompt compared to humans.

5.2 Prompt grounding with turns

Compared to D3ST, D3ST-Turn achieves absolute gains of 13% and 6.1% on SGD-X and SGD, respectively (#1 vs #5, Table 3). D3ST + SGD-X DA outperforms D3ST-Turn on the seen services because it has been trained with these prompt paraphrases, whereas our model *does not* see these prompts during training. Our model generalises more robustly as demonstrated by the 1.1% (#4 vs #5) JGA improvement on unseen SGD-X services.

Our results show that grounding the model in knowledge-seeking turns, communicated before a slot is mentioned, addresses weaknesses of data augmentation (DA) with synthetic prompts. Such turns reflect how humans use the language in conversation when they communicate slot values, and may help the model more readily identify the relevant context for value extraction. This approach generalises well to unseen domains and is robust: we outperform all baselines on SGD and closely match the performance of augmentation with human-written paraphrases on SGD-X.

Descriptions and knowledge-seeking turns are complimentary: the latter can be thought of as an *example* that could help the model interpret descriptions unseen at training time. Concretely, consider the *messaging* domain, unseen in training. To extract the name of a location sharing recipient, a model evaluated on SGD-X (*v5*) is prompted with the description *Name from address book*. Because T5 is pre-trained in a self-supervised way, without domain-specific finetuning, it may fail to identify that the aforementioned description refers to the name of a person: *address book* never appears in the SGD training corpus. By attending over *Who is the sharing recipient* and the description jointly, the model could interpret the description as referring to a person name. During training, the model has learned to identify names, for example, when predicting the value of the slot *stylist name*. Indeed, the D3ST-Turn JGA in this domain is 41.8% while the oracle model achieves just 28.5%. Our positive results may thus arise due to knowledge-seeking turns facilitating knowledge sharing between slots seen in training and unseen ones.

5.3 Prompt grounding with turns and slots

D3ST-TurnSlot achieves a 2.5% gain on SGD-X compared to D3ST-Turn, outperforming the human-written prompts (#4 vs #5, Table 3). We posit that this is due to the high quality annotations SGD-X provides. This hypothesis is motivated by our empirical observation that slot names in SGD-X can contain more information compared to SGD ones. For example, the slot *private visibility* in SGD is annotated as *private visibility yes or no* in SGD-X (*v5*), which cues the model on which values should be generated for this slot. Also, in SGD-X, the slot names and descriptions may be complimentary. For example, the slot *clock time of alarm* is described as *Time for which the alarm is set* (SGD-X), whereas in SGD the equivalent slot name, *alarm time*, is described as *Time of the alarm*. On its own, the SGD description could refer to both an alarm to be created or an existing alarm, whereas the SGD-X description unambiguously identifies the slot as referring to an existing alarm.

D3ST-TurnSlot outperforms D3ST + SGD-X DA by 0.9% on SGD (#4 vs #6, Table 3) but lags behind D3ST-Turn by 1.1%. This confirms our earlier observation that, in SGD, unlike in SGD-X, slot names may not provide information about the slot that is not already contained in the description. We also find that there are slot name ambiguities across the SGD train and test sets. For example, the *location* slot in the training set refers to cities, whereas in the test set it refers to addresses. This finding correlates with the study of Zhao et al. (2022), the D3ST authors, who find that lack of information in slot names and ambiguity lead to degraded JGA (on both SGD and SGD-X) of slot-name driven models compared to D3ST¹⁰. Our positive results on SGD-X show that combining the two sources of information can improve model robustness if they are complimentary and unambiguous.

5.4 Grounded prompt ensembling

We apply GPE by running three inference calls with distinct but semantically equivalent grounded prompts. We take the most common generation as the prediction. Table 5 shows significantly improved robustness compared to single-prompt decoding. Interestingly, D3ST-TurnSlot is improved

¹⁰A slot-driven model uses slot names instead of descriptions. For our example in Figure 1 the equivalent prompt is *0=name, 1=city, i1) get weather*. We refer the reader to Sections 4.3, 4.4 and 4.6 in Zhao et al. (2022) for detailed comparisons of the effectiveness of these competing approaches.

Model	SGD	SGD-X	Seen	Unseen	SS ↓
D3ST-Turn	77.2 1.4	71.7 2.2	90.8 2.3	65.4 2.2	28.1 8.5
D3ST-TurnSlot	75.0 0.3	72.8 0.8	91.4 0.7	66.6 1.0	19.2 4.5

Table 5: GPE improves SGD/SGD-X performance. Faded numbers are absolute improvements relative to the single pass models in Table 3 in rows # 5 & # 6.

Model	SGD	SGD-X	Seen	Unseen	SS ↓
T5DST	70.0	50.4	58.5	47.7	87.0
MT-SGDST	80.1	60.8	72.5	56.9	69.5
SDT-Seq	76.3	-	-	-	-
SDT-Ind	78.2	-	-	-	-
D3ST-Turn (ours)	75.8	69.5	88.5	63.2	36.6
D3ST-TurnSlot (ours)	74.7	72.0	90.7	65.6	23.7

Table 6: SOTA DST models on SGD and SGD-X. Bottom rows repeated from Table 3 for easy comparisons.

less compared to D3ST-Turn on both SGD and SGD-X owing to its significantly smaller prompt sensitivity (23.7 compared to 36.6, Table 3, # 5 vs #6). This shows that slot names increase the confidence of the model in its predictions, which may explain why we found D3ST-TurnSlot to slightly outperform D3ST-Turn (Section 5.3).

5.5 Comparison with other models

We compare D3ST-Turn/TurnSlot with SOTA DST models (Table 6). **T5DST** (Lee et al., 2022a) generates a slot value when prompted with a dialogue concatenated with a single description. The state is predicted *iteratively* by prompting the model with each description. **MT-SGDST** (Kapelonis et al., 2022) uses semantic annotations, state history and handcrafted features with a multi-head BERT model for iterative prediction. **SDT-Seq** (Gupta et al., 2022) grounds the state tracker in a prompt containing a dialogue and its target state sequence and, like D3ST, predicts the entire state in a single pass. **SDT-Ind** is an iterative version of SDT-seq, using annotated turns as prompts.

Grounding prompts in knowledge-seeking turns makes D3ST competitive with SOTA approaches, significantly outperforming T5DST. MT-SGDST is better on the SGD but degrades significantly on SGD-X. Because it uses state histories and semantic annotations instead of system turns, this model suffers from large performance variability (Appendix B.2): the difference between max and min SGD-X JGA across three runs is 11.8% for this model but just 1.1% for D3ST-Turn. While not fully closing the gap on the SGD benchmark, we demonstrate comparable performance and significantly improved robustness.

Model	SGD	SGD-X	Seen	Unseen	SS ↓
D3ST	76.0	69.2	86.8	63.3	38.5
D3ST + SGD-X DA	77.4	75.6	93.2	69.7	19.8
D3ST-TurnSlot (ours)	77.4	76.0	92.6	70.5	20.5

Table 7: Prompt grounding improves T5-large D3ST.

D3ST-Turn achieves 75.8% on SGD, which is comparable with SDT-Seq (76.3%). Our model is faster to train and decode owing to reduced prompt lengths and predicting a shorter state sequence¹¹. SDT-Ind is better because it is prompted to return the value of each slot iteratively, with an example of how that typical slot occurs in conversation. GPE is cheaper and reduces the performance gap between SDT-Ind and D3ST-Turn to just 1.0%.

In terms of human effort, our approach is more scalable than, or comparable to, recent work SDT uses entire annotated dialogues or annotated turns as prompts (Figure 1, Gupta et al. (2022)). These are defined by developers for unseen services, which is comparable to writing turns for each slot. Zero-shot transfer learning (Campagna et al., 2020) requires knowledge-seeking turns for generating synthetic dialogues¹² used to bootstrap DST models for new services. However, constraining entire dialogue generation is non-trivial and handcrafted grammars are required for each domain. This is very difficult to apply to the setting we consider, due to the large number of domains and complex multi-domain dialogue flows. We show that robust generalisation to new services can be achieved with few knowledge-seeking turns per slot which can be selected from the training corpus during finetuning and written by the developers for new services.

5.6 Scaling behaviour

Larger models have enhanced language understanding and common sense knowledge (Raffel et al., 2020; Zhou et al., 2021), reflected in the 12.7% improvement of the baseline T5-large D3ST performance on SGD-X compared to its T5-base counterpart (#1, Tables 3 and 7). Exposing the model to diverse prompts is still important, as demonstrated by the improved JGA of D3ST + SGD-X DA. We find that D3ST-TurnSlot matches the SGD performance and achieves a slight improvement on SGD-X (0.6%), demonstrating that our approach scales to larger language models.

¹¹SDT predicts all slots, including inactive ones.

¹²The *direct questions* defined by Campagna et al. (2020) are KSTs. See examples here: <https://bit.ly/3yvgGqi>.

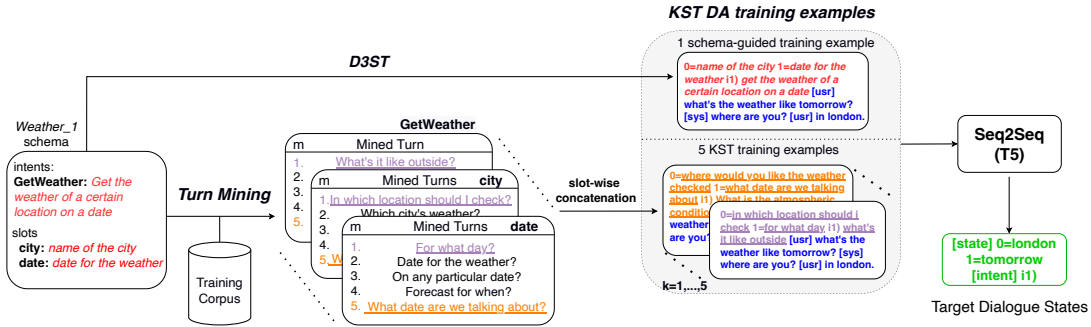


Figure 3: Data preprocessing pipeline for KST augmentation. KST training examples share the conversation with the schema-guided training example, and concatenated KSTs, underlined, replace the schema-guided prompt.

#	Decoding Prompt	SGD	SGD-X	Seen	Unseen	SS ↓
1	Turn [GPE]	74.9 [76.7]	71.7 [73.9]	91.9 [92.1]	65.0 [67.7]	30.7 [22.6]
2	TurnSlot [GPE]	73.8 [75.3]	71.0 [72.9]	91.0 [91.7]	64.3 [66.6]	31.2 [23.1]
3	D3ST	74.4	66.7	88.8	59.4	43.4

Table 8: JGA of D3ST with KST augmentation decoded with different prompt formats. GPE further improves these models (improvements inside brackets).

5.7 Data augmentation or prompt grounding?

In Section 5.2, we discussed that knowledge-seeking turns may facilitate knowledge sharing between seen and unseen slots. We now investigate whether jointly encoding the turn and description by including them in the same prompt is the only way to impart this property or whether this can be achieved by augmenting the training data with prompts containing *only* knowledge-seeking turns.

Figure 3 shows our experimental setup for augmentation with knowledge-seeking turns. We sort the mined turn lists for each slot from Section 3.2 according to their Jaccard distance to the corresponding SGD schema description. We create $k = 5$ increasingly diverse prompts by replacing all the slot descriptions in a schema-guided training example with the k th knowledge-seeking turn. The resulting finetuning set is the same size as SGD-X.

Comparing the performance of augmented and grounded models (#5 & #6, Table 3 vs #1 & #2, Table 8) shows that grounding D3ST is slightly more effective on SGD.

On SGD-X, decoding the KST-augmented D3ST with TurnSlot prompt format causes a small (1%) regression with respect to D3ST-TurnSlot, possibly due to mismatch between the train and test prompt formats. "Turn" decoding slightly improves over D3ST-Turn. Hence, both grounding and data augmentation with knowledge-seeking turns are effective for robust DST. Training with augmented data, converges slower and is resource intensive. More-

#	Model	SGD	SGD-X	Seen	Unseen	SS ↓
1	D3ST-TurnSlot	77.4	76.0	92.6	70.5	20.5
2	D3ST + KST DA/D3ST	76.3	72.8	92.5	66.3	26.0
3	D3ST + KST DA/Turn	76.1	74.6	93.4	68.4	26.0
4	D3ST + KST DA/TurnSlot	75.8	73.6	92.2	67.4	28.2

Table 9: Grounding prompts (#1) is more effective compared to KST-augmentation (#2 - #4) for robust DST with T5-large (770M parameters).

over, we find that grounding is more effective for larger language models (Table 9).

5.8 Why is grounding more effective?

Decoding the KST-augmented model with the SGD/SGD-X schemata alone (i.e., without grounding) leads to a decrease in the unseen performance (#1 & #2 vs #3, Table 8). Grounding the prompt in KSTs at decoding time is crucial for improved robustness. As discussed in Section 5.2, these turns facilitate knowledge sharing between seen and unseen slots. Without them, the model cannot access knowledge encoded in its weights, and robustly predict the dialogue state when the prompts are too dissimilar to the training schemata (Table 10).

SGD	SGD-X (avg)	v1	v2	v3	v4	v5
0.43	5.03	1.07	0.67	3.35	11.3	8.76

Table 10: JGA difference between KST-augmented models decoded with Turn and D3ST prompt formats (#1 & #3, Table 8). SGD-X JGA broken down by variant. $v5$ schema is the most dissimilar to the SGD test schema.

6 Conclusion

Grounding D3ST and data augmentation with knowledge-seeking turns are effective for robust schema-guided DST. Both improve D3ST robustness by a large margin compared to strong baselines and yield similar benefits as training on large,

diverse collection of human-written prompts. Our proposed approach is competitive with or outperforms other SOTA DST models on SGD and SGD-X. We have also showed how prompt engineering can be applied to boost model robustness through grounded prompt ensembling, a novel technique that uses a single model for ensembling.

7 Limitations

One limitation of our approach is our decision to select the turns from the training data manually rather than automatically. Selecting k -diverse turns automatically is possible but requires efficient implementations given the size of the corpus and the quadratic complexity of the naive algorithm in the number of candidate turns. Implementing such algorithms requires far more expertise and time commitment compared to ensuring the selected turns are diverse manually. Such an approach is described by Lee et al. (2022b). However, in a follow-up study, we have confirmed the generality of our approach by replicating our experiments with large-language model generated data or sampling randomly from large dialogue corpora during the first epoch of training. This allays concerns regarding the vulnerability of our method to human bias.

While not an issue for SGD or other large scale corpora, the diversity of the training corpus may influence the performance of our approach as extracting lower diversity turns is expected to limit robustness improvements. However, knowledge-seeking turns existing in small corpora can be used to query large, possibly unlabeled, conversational databases to ensure prompt diversity. We left a detailed study of the impact of prompt diversity to DST robustness to future work.

Finally, for practically implementing our approach for unseen services, we require the developers to provide few examples of knowledge-seeking turns. Our currently in progress work explores generation of such turns automatically with very large language models.

Acknowledgements

Alexandru Coca was supported supported by EP-SRC grant EP/R513180/1. He would like to acknowledge Harrison Lee and Raghav Gupta from Google Research for support and guidance with D3ST and T5DST implementation. Additionally, he would like to acknowledge the helpful feedback

and suggestions that shaped this work from Matt Henderson (reka.ai). Weizhe Lin was supported by a Research Studentship funded by Toyota Motor Europe (RG92562(24020)). We also thank Howard Mei from University of Cambridge for help with editing the final draft. Authors would like to acknowledge the improvement suggestions made by anonymous reviewers and the SIGDIAL program committee.

References

- Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S. Lam. 2020. [Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 122–132. Association for Computational Linguistics.
- Jie Cao and Yi Zhang. 2021. [A comparative study on schema-guided dialogue state tracking](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 782–796. Association for Computational Linguistics.
- Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. [Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3002–3017. Association for Computational Linguistics.
- Alexandru Coca, Tseng Bo-Hsiang, Lin Weizhe, and Bill Byrne. 2022. [Improving generalisation of schema-guided dialogue state tracking via tree-based paraphrase ranking](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 527–533, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, Dilek Hakkani-Tur, and Amazon Alexa AI. 2019. [Dialog state tracking: A neural reading comprehension approach](#). In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 264.
- Raghav Gupta, Harrison Lee, Jeffrey Zhao, Yuan Cao, Abhinav Rastogi, and Yonghui Wu. 2022. [Show, don't tell: Demonstrations outperform descriptions for schema-guided task-oriented dialogue](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*,

- pages 4541–4549. Association for Computational Linguistics.
- Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.
- Shuo Huang, Zhuang Li, Lizhen Qu, and Lei Pan. 2021. [On robustness of neural semantic parsers](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 3333–3342. Association for Computational Linguistics.
- Léo Jacqmin, Lina M. Rojas Barahona, and Benoit Favre. 2022. [“do you follow me?”: A survey of recent approaches in dialogue state tracking](#). In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 336–350, Edinburgh, UK. Association for Computational Linguistics.
- Eleftherios Kapelonis, Efthymios Georgiou, and Alexandros Potamianos. 2022. [A multi-task BERT model for schema-guided dialogue state tracking](#). In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 2733–2737. ISCA.
- Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. [Dialogue state tracking with a language model using schema-driven prompting](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4937–4949. Association for Computational Linguistics.
- Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2022a. [SGD-X: A benchmark for robust generalization in schema-guided dialogue systems](#). In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10938–10946. AAAI Press.
- Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022b. [Deduplicating training data makes language models better](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8424–8445. Association for Computational Linguistics.
- Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Zhenpeng Zhou, Paul A. Crook, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. 2021a. [Zero-shot dialogue state tracking via cross-task transfer](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7890–7900. Association for Computational Linguistics.
- Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021b. [Leveraging slot descriptions for zero-shot cross-domain dialogue StateTracking](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5640–5648, Online. Association for Computational Linguistics.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. [Rethinking the role of demonstrations: What makes in-context learning work?](#) In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics.
- Nikola Mrksic, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve J. Young. 2017. [Neural belief tracker: Data-driven dialogue state tracking](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1777–1788. Association for Computational Linguistics.
- Shashi Narayan, Gonçalo Simões, Yao Zhao, Joshua Maynez, Dipanjan Das, Michael Collins, and Mirella Lapata. 2022. [A well-composed text is half done! composition sampling for diverse conditional generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1319–1339, Dublin, Ireland. Association for Computational Linguistics.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. [SOLOIST: few-shot task-oriented dialog with A single pre-trained auto-regressive model](#). *CoRR*, abs/2005.05298.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. [Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*,

AAAI 2020, *The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8689–8696. AAAI Press.

Stephanie Seneff and Joseph Polifroni. 2000. Dialogue management in the mercury flight reservation system. In *ANLP-NAACL 2000 Workshop: Conversational Systems*.

Jason W. Wei and Kai Zou. 2019. [EDA: easy data augmentation techniques for boosting performance on text classification tasks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6381–6387. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [Huggingface’s transformers: State-of-the-art natural language processing](#). *CoRR*, abs/1910.03771.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Dian Yu, Mingqiu Wang, Yuan Cao, Izhak Shafran, Laurent El Shafey, and Hagen Soltau. 2022. [Un-supervised slot schema induction for task-oriented dialog](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1174–1193. Association for Computational Linguistics.

Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. [Description-driven task-oriented dialog modeling](#). *CoRR*, abs/2201.08904.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, and Xiang Ren. 2021. [Pre-training text-to-text transformers for concept-centric common sense](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

A Turn mining details

For 31 out of the 214 slots there are no knowledge-seeking turns or there are less than 5 distinct knowl-

edge seeking turns¹³ in the dialogue corpus. These includes *result* slots which are communicate by the agent upon user query, such as the name or time of an existing alarm in the *Alarms_1* service. These slots do not appear in state annotations. Moreover, SGD dialogue flows are generated by semantic-level interaction between two machines modelled using push-down automata (Rastogi et al., 2020). As such, not all dialogue flows are covered. For example, in the *Alarm_1* service the user always states the name of a new alarm and the time they want to set it for so the system never asks for what time the new alarm should be set.

We circumvent these issues with two simple strategies, which are applied depending on whether a slot has knowledge-seeking turns in other services or not. The majority of the slots fall in the former case.

Turn copy The *only* knowledge seeking turn for the *fare* slot in *Buses_1* service is *Thanks for that, how much did it cost?*. However, price is a generic concept which appears in other services (e.g. *Events_1*) so instead of reducing prompt diversity by always using this turn, we copy knowledge-seeking turns from other services. In this instance, *How much did it cost?*, *Ticket fare for each passenger?*, *Price per ticket?* and *What price?* are copied. This strategy is applied to all slots that appear in other services.

Span selection For just 8 slots, a relevant span appearing before or after the slot value is selected from turns annotated with actions $\text{INFORM}(s=v)$ or $\text{CONFIRM}(s=v)$ where s is a slot for which no knowledge-seeking turns exist and v is its value. For example, there are no turns where the system or user ask for the seating class of an airline ticket. We select the span *class flight ticket* instead of a full turn from the system turn *Please confirm an Economy class flight ticket to NY, tomorrow.*. The semantic annotation of this turn is $\text{CONFIRM}(\text{destination}=\text{NY}), \text{CONFIRM}(\text{date}=\text{tomorrow}), \text{CONFIRM}(\text{seating_class}=\text{Economy}), \text{CONFIRM}(\text{passengers}=1)$.

B Experimental details

B.1 D3ST implementation

We process the data as described by Zhao et al. (2022) with the following differences, which were

¹³Only 25 of these slots are unique as some slots repeat across services.

indicated by the paper authors upon private communication: (1) the indices are separated by the = symbol in both the inputs and the targets, (2) for categorical slots which take the *dontcare* special value, our output contains *slot_index: dontcare* substring and we do not include the special value in the prefix and (3) we lowercase inputs and targets.

The examples are truncated to the last 1,024 tokens on the input side for the baseline and discarded altogether for Turn/TurnSlot prompt formats¹⁴. We optimise with the Adafactor optimizer and effective batch size 32, starting from the initial weights `google/t5-v1_1-base` published by huggingface (Wolf et al., 2019). We interpolate the learning rate linearly between 0 and 10^{-4} over the first 1000 steps and keep it constant thereafter. We select the model by evaluating the development set joint goal accuracy (JGA) every 5000 gradient updates, stopping the training if said metric fails to improve after 3 consecutive evaluations.

All results in Section 5 are averages of three runs initialised with different random seeds. For all experiments, we used the same hyperparameters and stopping criteria as just described, with the exception of training the D3ST + SGD-X DA and D3ST + KST DA experiments for T5-large (rows 2 in Table 7 and last three lines in 9) where we allow all runs 1 epoch of augmented data (each SGD conversation is seen 6 times) due to limited computational budget.

B.2 MT-SGD implementation

The numbers presented are averages of three runs. The first SGD run (JGA of 83.2%) is based upon a metric file received from the authors. We could only reproduce 82.7% of the quoted number, but we include the higher number in our average. We trained the model using the publicly available code¹⁵ twice more obtaining to obtain 77% and 80.2%. On SGD-X the JGA range is between 54.6% and 66.4% across three runs. We selected the best checkpoint as indicated in the repository’s instructions.

C Backtranslation experiment

We experiment with larger backtranslation datasets to see if finetuning D3ST on a dataset the same size as the SGD-X dataset (Section 4.2) can improve results. We created two more variants by backtrans-

¹⁴This is just around 0.05% of the data.

¹⁵See it here: bit.ly/3j8sPwj

Size	SGD	SGD-X	Seen	Unseen	SS
4x	72.1	62.2	84.0	54.9	53.1
6x	71.5	61.0	82.5	53.8	54.4

Table 11: SGD and SGD-X JGA with backtranslation datasets of different size. We repeat line 2 from Table 3 in the top row, for easy comparison

Metric	Method	v1	v2	v3	v4	v5
BLEU	Backtranslation 4x	36.4	26.01	18.9	-	-
	Backtranslation 6x	51.3	37.2	29.5	23.4	18.2
self-BLEU	Backtranslation 4x	-	49.3	41.7	-	-
	Backtranslation 6x	-	55.3	49.7	44.6	39.6

Table 12: Lexical diversity metrics of backtranslated prompts. self-BLEU measures diversity of n sentences

lating the SGD schema via French and Russian, as done by Huang et al. (2021).

Augmenting with these additional examples negatively impacts model robustness (Table 11). This may arise because increasing the number of training examples significantly (Table 12) does not increase the prompt diversity by a large margin, and so the training distribution of the prompts is closer to the training data. Creating a diverse collection of paraphrases via backtranslation is thus challenging, as it requires access to translation systems to high-difficulty languages. This is necessary, since, as shown in Table 12 (BLEU, column 2) translating to high-resourced languages such as French yields paraphrases that are lexically more similar to the input and are not as effective in improving the model robustness. Meanwhile, translation to difficult languages leads to semantic errors which may harm DST. For example, *Station where the bus is leaving from* is backtranslated to *Bus departure/arrival station* and *Station where the bus is going to* is backtranslated as *bus station* (via Japanese).

By grounding the model in turns collected from the corpus, not only do we create diverse inputs, but we guarantee that these correctly represent fine grained semantics and by-pass the issues encountered when constructing prompts via paraphrasing.

D SGD results

In the main body we report the SGD JGA accuracy as an upper bound for the D3ST model robust accuracy. To make our tables readable, we do not include SGD performance breakdown by seen/unseen services in Section 5. We include it in Table D to facilitate future comparisons

Model	SGD	SGD-Seen	SGD-Unseen
D3ST	69.8	92.8	62.2
D3ST + SGD-X DA	73.8	92.7	67.5
D3ST-Turn	75.8	92.9	70.1
D3ST-TurnSlot	74.7	92.8	68.7
D3ST + KST DA/Turn	74.9	92.6	69.0
D3ST + KST DA/TurnSlot	73.8	92.5	67.6
D3ST + KST DA/D3ST	74.4	92.8	68.3

Table 13: Breakdown on SGD JGA into seen and unseen services JGA for models reported Tables 3 and 8.